

Japanese Unexamined Patent Application Publication No. 6-044188

[Title of the Invention] SCHEDULER

[0014] The gate array 22 is an interface logic circuit which controls the keyboard 3, the liquid crystal display unit 7, the printer section 9, etc. The FDC 24 is a logic circuit which controls the FDD section 5. The schedule data processing operation conducted when the blank capacity of the RAM 14 storing schedule data becomes short will now be described with reference to the flowcharts shown in Figs. 3 and 4.

[0015] When the operator of the electronic typewriter 1 prepares schedule data anew, the CPU 10 performs checking of the balance of the internal memory of the RAM 14 (step S1, hereinafter step S1 is simply referred to as S1, this applying also to the other steps). When there is a blank memory space larger than a record (a date and a piece of schedule information corresponding to such a date) in the RAM 14 (S1: Yes), schedule data are prepared on the basis of the entered data (S12). If there is not a blank memory space for a record in the RAM 14 (S1: No), checking is conducted as to whether or not there are data already implemented (S2). Whether or not data have been implemented is checked by seeing whether or not an

unnecessity information is added to the stored schedule data. As shown in the conventional case, addition of unnecessity information is accomplished by adding a mark V to schedule data becoming unnecessary which were to be implemented. A date and time data counted by the RTC 16 is attached as a time stamp to the schedule data bearing a mark V.

[0016] When there is no schedule data bearing the mark V added thereto, i.e., when there is no implemented data (S2: No), a message, for example, "There is no deletable data" is displayed on the liquid crystal display unit 7 (S11), thereby ending the processing. When there are implemented data (S2: Yes), the implemented data are displayed on the liquid crystal display unit 7, and simultaneously, a message "May data be deleted?" is displayed to ask the user whether or not to delete the data (S3). When a user's instruction to delete the implemented data is keyed in from the keyboard 3 (S3: Yes), a message expressing whether or not to retain the implemented data to be deleted in a floppy disk 11 of the FDD section 5 is displayed to ask the user to acknowledge (S4).

[0017] If an instruction for not retaining is entered from the keyboard 3 (S4:No), the processing comes to an end by deleting the implemented data from the RAM 14 (S10). If an instruction to retain implemented data is entered (S4: Yes), the balance of memory of the

floppy disk 11 provided in the FDD section 5 is checked up (S5).

[0018] When the blank capacity of the floppy disk is too small to retain the implemented data (S5: NG), an error message such as "The memory capacity capable of retaining data is unavailable" is displayed (S13). The data storing step is skipped, and the implemented data are deleted (S10), thereby ending the processing. If the memory balance of the floppy disk 11 of the FDD section 5 is sufficient to permit storage of the implemented data (S5: OK), opening processing of the file is executed (S6). The file opening processing means a processing of opening a file bearing a specified file name. If, for example, the file name of the implemented data is assumed to be "CHECK", a file having the file name "CHECK" is opened. Then, the file ID and the file control information (number of registered schedule data) are written into the floppy disk 11 by means of the FDD section 5 (S7), and the implemented data are written into the floppy disk 11 (S8). When information necessary for the file is written in, the file closing processing is executed (S9). This represents a closing processing of a file opened by the file opening processing (S6). Then, the implemented data are deleted from the RAM 14 (S10), thereby ending the processing.

[0019] The procedure of executed processing through a process of deleting the implemented data from the RAM 14, shown in Fig. 3,

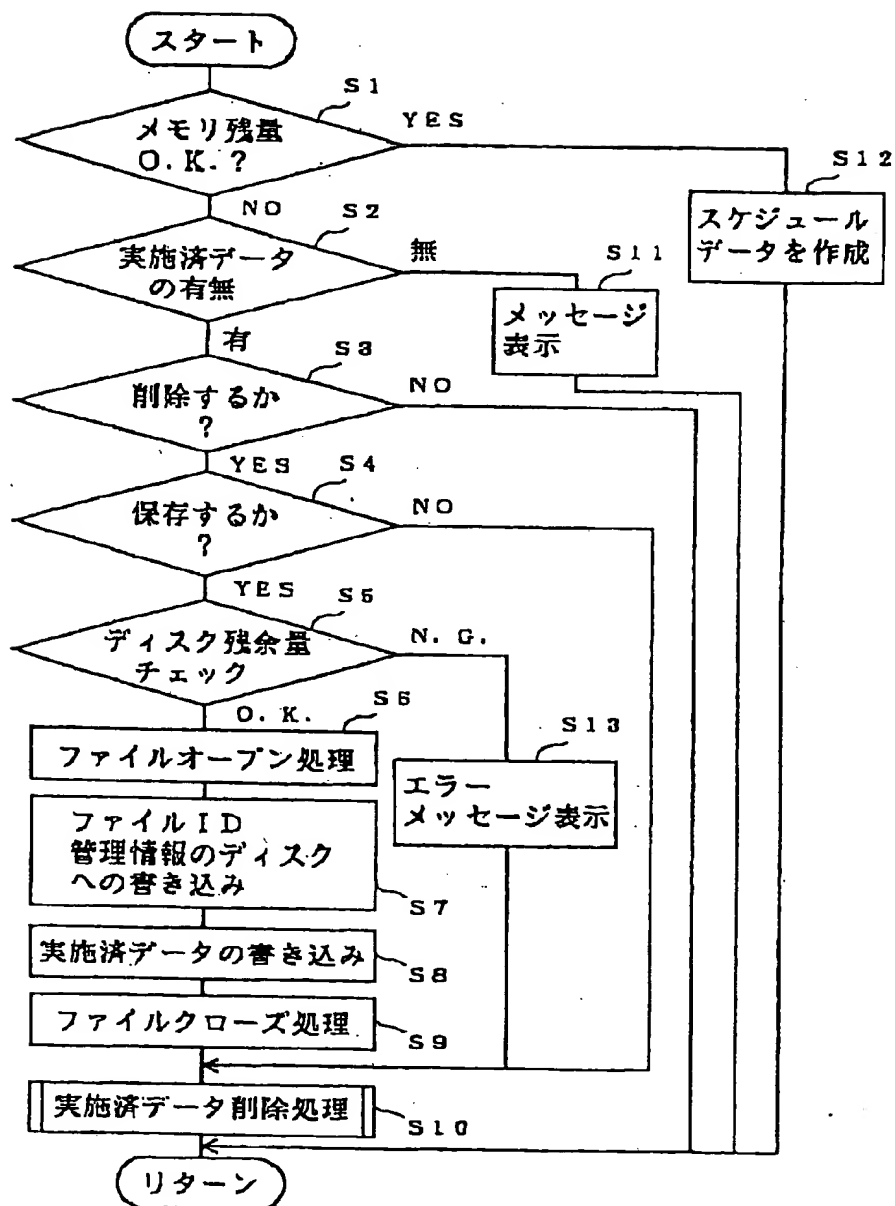
will now be described with reference to the flowchart shown in Fig. 4. First, it is checked whether or not there are schedule data in the RAM 14 (S20). When the RAM 14 contains no schedule data (S20: No), nothing is done since there is no data to be deleted, thereby ending the step. When schedule data are present (S20: Yes), the maximum value of record numbers is set in the record number I serving as an index (S21). Then, it is checked whether or not the schedule data of the record number set in I are implemented data (S22).

[0020] Checking as to whether or not data are implemented data is accomplished by seeing whether or not a mark V or a time stamp serving as unnecessary information is affixed to the schedule data as described above. When the thus checked data are not implemented data (S22: No), the value of I is reduced by 1, without deleting the schedule data, and the schedule data of the record number represented by the resultant value are checked (S26, S27 and S22). If the schedule data represented by I are implemented data (S22: Yes), it is checked whether or not the schedule data are the final record (S23).

[0021] Being the final record is determined by seeing whether or not the value of I is equal to the maximum value of the record numbers. If the final record (S22: Yes), the schedule data in a number corresponding to a record are erased (S25). When the

schedule data represented by I are not the final record (S23: No), the schedule data are erased, and then, sliding processing of the data subsequent to the erased data is executed (S24). Sliding processing is an operation of re-storing schedule data to be stored in the RAM 14 so that a vacancy is not produced in the address sequence. That is, the schedule data stored in the next address is stored in the address in which the erased schedule data have been stored. The schedule data stored in the next next address is stored in the next address. These steps are repeated until the final record. After executing sliding process, the schedule data of the address having formed the final record prior to execution of sliding processing is erased (S25). When deleting the schedule data corresponding to one record as described above, the record number I is reduced by one (S26), and the steps S22 to S26 are repeatedly executed until I becomes zero (S27). Implemented data are thus deleted from the RAM 14.

[Fig. 3]



[Fig. 3]

Start

S1: Memory balance is OK?

S2: Presence of implemented data

Yes

S3: Delete?

S4: Is it stored?

S5: Check disk balance

S6: File opening processing

S7: Write file ID control information in disk

S8: Write in implemented data

S9: File closing processing

S10: Delete implemented data

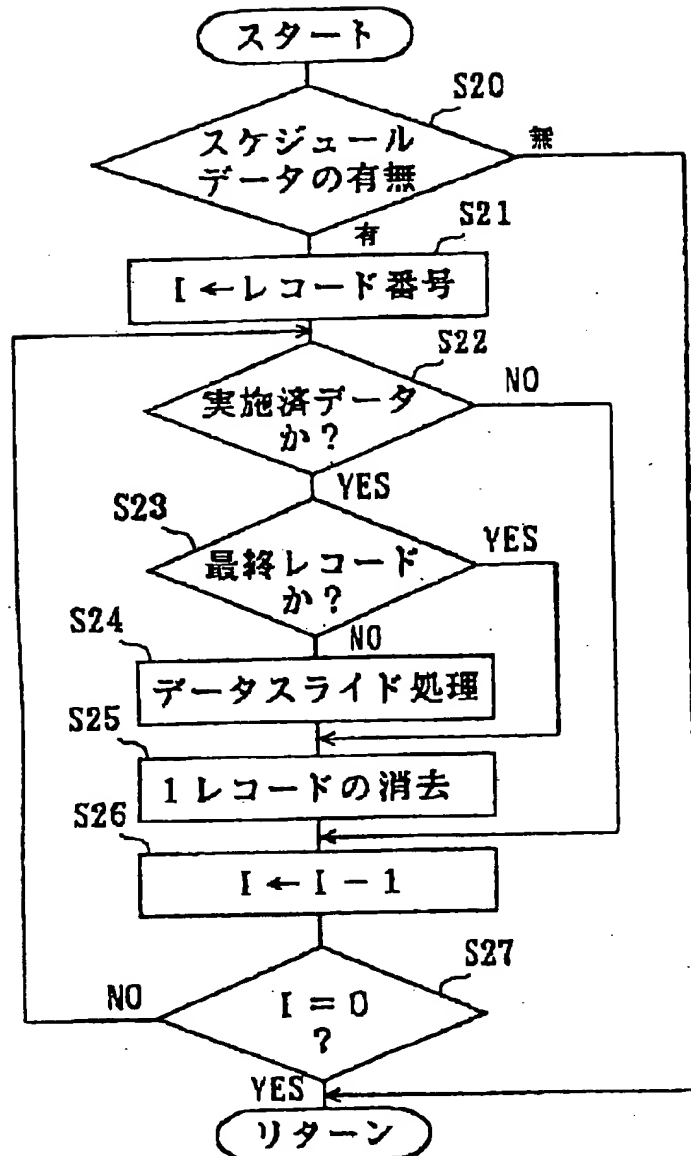
Return

S11: Display message

S12: Prepare schedule data

S13: Display error message

[Fig. 4]





[Fig. 4]

Start

S20: Presence of schedule data

Yes, No

S21:  $I \leftarrow$  Record number

S22: Implemented data?

S23: Final record?

S24: Data sliding processing

S25: Erase one record

Return

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平 6 - 4 4 1 8 8

(43) 公開日 平成6年(1994)2月18日

(51) Int. Cl. <sup>5</sup>

識別記号

庁内整理番号

F I

技術表示箇所

G 0 6 F 15/02  
15/21

3 5 5 A 7343-5 L  
L 7052-5 L

審査請求 未請求 請求項の数 2

(全 7 頁)

(21) 出願番号 特願平4-199903

(22) 出願日 平成4年(1992)7月27日

(71) 出願人 000005267

ブラザー工業株式会社

愛知県名古屋市長区瑞穂区苗代町15番1号

(72) 発明者 古嶋 時人

愛知県名古屋市長区瑞穂区苗代町15番1号

ブラザー工業株式会社内

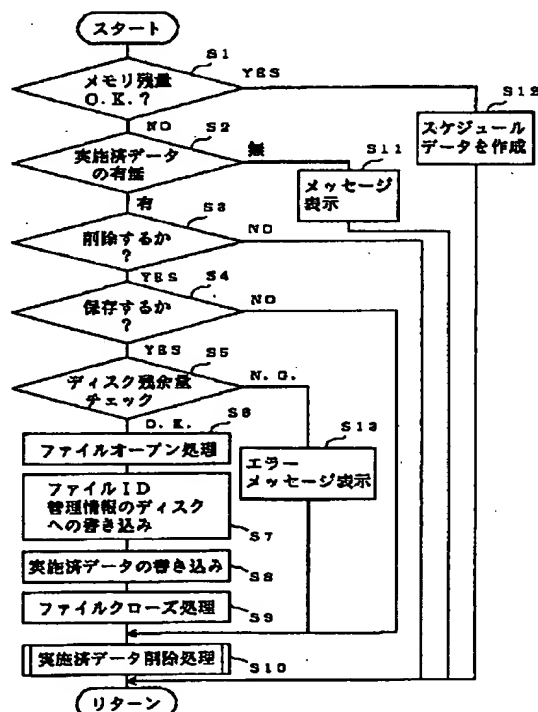
(74) 代理人 弁理士 足立 勉

(54) 【発明の名称】 スケジューラ

(57) 【要約】

【目的】 実施済みの不用データの管理にてユーザが煩雑な手間を加える必要のない使い勝手のよいスケジューラを提供すること。

【構成】 RAMに1レコード分以上の空容量があれば (S1: YES) スケジュールデータが作成される (S1.2) が、1レコード分の空容量がなければ (S1: NO)、実施済データの有無がチェックされ (S2)、実施済データが存在し (S2: YES)、その実施済データを削除するならば (S3: YES)、削除しようとする実施済データをフロッピーディスクに保存するかどうかのメッセージを表示してユーザに確認させる (S4)。実施済データを保存するならば (S4: YES)、その実施済データをフロッピーディスクに保存する (S6~S9) と共に実施済データをRAMから消去する (S10)。



## 【特許請求の範囲】

【請求項1】 日時と該日時に対応した情報とからなる予定情報を入力する入力手段と、

上記入力手段により入力される予定情報を格納する第1記憶手段と、

上記第1記憶手段に格納される予定情報を外部に出力する出力手段と、

上記記憶手段に格納される予定情報を削除する際、該削除した予定情報を格納する第2記憶手段とを備えたことを特徴とするスケジューラ。

【請求項2】 日時と該日時に対応した情報とからなる予定情報を入力する入力手段と、

上記入力手段により入力される予定情報を格納する記憶手段と、

上記記憶手段に格納される予定情報を外部に出力する出力手段と、

上記記憶手段に格納される予定情報の内、不要となった予定情報に対し外部から不要情報を付加する不要情報付加手段と、

上記記憶手段の未使用容量を検出し該検出した未使用容量が予め定められた値以下のとき、上記不要情報付加手段により不要情報が付加された予定情報を上記出力手段から出力するメモリ制御手段とを備えたことを特徴とするスケジューラ。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】本発明は、電子手帳やワープロ等に搭載され、日時とその日時に応じた情報を管理するスケジューラに関する。

## 【0002】

【従来の技術】従来、電子手帳やワープロ等に搭載されるスケジューラでは、日時とその日時に対応した情報（以下スケジュールデータと呼ぶ）が記憶されるメモリの容量はユーザが管理していた。例えば図5に示すスケジューラの場合、チェック欄にVマークの付加されたスケジュールデータが実施済みのスケジュールデータであり不必要なデータとなる。このスケジューラでは、十分なメモリの容量が残っていないことを、ユーザが新しくスケジュールデータを入力できないことで初めて知ることになる。新しくスケジュールデータを入力できない場合は、ユーザはVマークの付加されたスケジュールデータ（実施済データ）から不要なスケジュールデータを選択し、削除して、新しいスケジュールデータを入力する。

## 【0003】

【発明が解決しようとする課題】しかしながらこのような従来のスケジューラでは、一旦メモリから削除されてしまったスケジュールデータは、ユーザが二度と確認することができない。そのため実施済みのスケジュールデータであっても、後から確認する必要性が生じたとき、

または誤って必要なスケジュールデータを削除してしまったとき等、削除した実施済データはどこにも格納されておらず確認することができなかった。

【0004】またユーザは、スケジュールデータを入力する度に、スケジュールデータ格納されるメモリの容量を常に気にしていなければならなかった。そして新しいスケジュールデータが入力できないことで初めてメモリの容量が足りないことを知らされ、実施済みのデータの内容を検討して、該当するスケジュールデータのみを削除しなければならなかった。特に、急いで新しいスケジュールデータを入力した場合や、多くのデータを入力したい場合等にメモリの容量が足りなくなってしまうと操作が煩わしく、誤って必要なスケジュールデータを削除してしまう可能性があった。

【0005】本発明は、上記の問題点を解決するためになされたものであり、実施済みの不要データの管理にてユーザが煩雑な手間を加える必要のない使い勝手のよいスケジューラを提供することを目的とする。

## 【0006】

20 【課題を解決するための手段】この目的を達成するためになされた請求項1記載の発明（以下第1発明とする）は、図6（a）に例示するように、日時と該日時に対応した情報とからなる予定情報を入力する入力手段と、上記入力手段により入力される予定情報を格納する第1記憶手段と、上記第1記憶手段に格納される予定情報を外部に出力する出力手段と、上記記憶手段に格納される予定情報を削除する際、該削除した予定情報を格納する第2記憶手段とを備えたことを特徴とするスケジューラを要旨とする。

30 【0007】請求項2記載の発明（以下第2発明とする）は、図6（b）に例示するように、日時と該日時に対応した情報とからなる予定情報を入力する入力手段と、上記入力手段により入力される予定情報を格納する記憶手段と、上記記憶手段に格納される予定情報を外部に出力する出力手段と、上記記憶手段に格納される予定情報の内、不要となった予定情報に対し外部から不要情報を付加する不要情報付加手段と、上記記憶手段の未使用容量を検出し該検出した未使用容量が予め定められた値以下のとき、上記不要情報付加手段により不要情報が付加された予定情報を上記出力手段から出力するメモリ制御手段とを備えたことを特徴とするスケジューラを要旨とする。

## 【0008】

40 【作用】第1発明のスケジューラでは、入力手段により入力されるスケジュールデータは、まず第1記憶手段に格納され必要に応じて出力手段より外部に出力される。第1記憶手段に格納されているスケジュールデータが削除されると、そのスケジュールデータは第2記憶手段に格納される。例えば第2記憶手段としてフロッピーディスク、ハードディスク等を用いることができ、第1記憶

手段から削除してしまったスケジュールデータでも、ユーザは第2記憶手段に格納されているスケジュールデータを確認することができる。

【0009】第2発明のスケジューラでは、入力手段により入力されるスケジュールデータは、記憶手段に格納され必要に応じて出力手段より外部に出力される。その際、不要となったスケジュールデータに対しては外部から不要情報付加手段により不要情報が付加される。メモリ制御手段は、記憶手段の未使用容量を検出しており、検出した未使用容量が予め定められた値以下の場合、不要情報が付加されたスケジュールデータが出力手段から出力される。ユーザは出力されたスケジュールデータから削除するか否かを決定するだけでよいことになる。そのため記憶手段の未使用容量をユーザが気にかける必要がない。

#### 【0010】

【実施例】以下、本発明を具体化した一実施例を図面を参照して説明する。図1は、本発明一実施例としてのスケジューラを搭載した電子タイプライタ1の全体図である。

【0011】電子タイプライタ1は、キーボード3、FDD（フロッピーディスクドライブ）部5、液晶表示装置7、プリンタ部9等から構成されている。キーボード3は、アルファベットや各種ファンクションキー等の複数のキーからなる周知のキーボードである。このキーボード3は、入力手段および第2発明の不要情報付加手段としての機能を果たしている。FDD部5は、周知のフロッピーディスク11へのデータの書き込みおよびフロッピーディスク11からのデータの読み出しを実行することにより内部メモリの有効利用を可能としている。FDD部5に装着されるフロッピーディスク11は、第1発明の第2記憶手段としての機能を果たしている。液晶表示装置7はスケジュールデータや文章等を表示するディスプレイ装置である。この液晶表示装置7は、出力手段としての機能を果たしている。プリンタ部9はキーボード3より入力したデータやFDD部5のフロッピーディスク11に格納されたデータを印字する印字装置である。

【0012】次に図2を参照して、電子タイプライタ1の電氣的構成を説明する。電子タイプライタ1は、制御部20、ゲートアレイ22、FDC（フロッピーディスクコントローラ）24等から構成され、これにキーボード3、液晶表示装置7、プリンタ部9等が接続されている。

【0013】制御部20は、CPU10、ROM12、RAM14、RTC（リアルタイムクロックチップ）16、およびこれら装置を接続するバス18等から構成される制御装置部である。CPU10は、ROM12に格納されているプログラム等に基づき、各種処理を実行する中央処理装置である。このCPU10は、第2発明の

メモリ制御手段としての機能を果たしている。ROM12は、後述するフローチャートを実行するためのプログラム等を格納するリードオンリメモリである。RAM14は、スケジュールデータ等を格納するランダムアクセスメモリである。このRAM14は、第1発明の第1記憶手段、第2発明の記憶手段としての機能を果たしている。RTC16は、計時手段としての機能を果たしており、年月日と曜日と時刻とを含むリアルタイムの時刻信号を出力する。バス18は、CPU10、ROM12、RAM14、RTC16等を接続するデータバスである。

【0014】ゲートアレイ22は、キーボード3、液晶表示装置7、プリンタ部9等を制御するインターフェース論理回路である。FDC24は、FDD部5を制御する論理回路である。以下、図3および図4のフローチャートを参照しつつ、スケジュールデータを格納するRAM14の空き容量が少なくなったときのスケジュールデータ処理動作を説明する。

【0015】電子タイプライタ1の操作を行う者が、新規にスケジュールデータを作成する際、CPU10によりRAM14の内部メモリ残量チェックが行われる（ステップS1、以下単にS1で表す。他のステップについても同じ）。RAM14内に1レコード（一つの、日時とその日時に対応した予定情報）分以上の空き容量がある場合（S1：Yes）は、入力されたデータに基づいてスケジュールデータを作成する（S12）。RAM14内に1レコード分の空き容量がない場合（S1：No）は、実施済データが有るかどうかのチェックを行う（S2）。実施済データであるかどうかのチェックは、格納されているスケジュールデータに対し不要情報が付加されているかどうかにより行われる。不要情報は従来例に示したように、操作を行うものが既に実施済み等で不要となったスケジュールデータに対しVマークを付加することで行われる。Vマークが付加されたスケジュールデータには、RTC16が計時する日時データがタイムスタンプとして付けられている。

【0016】ここでVマークが付加されたスケジュールデータがない、つまり実施済データが無い場合（S2：無）は、液晶表示装置7に「削除可能なデータがありません。」等のメッセージを表示（S11）して処理を終了する。実施済データが有る場合（S2：有）は、実施済データを液晶表示装置7に表示すると共に「削除してよろしいですか？」等のメッセージを表示して削除するかどうかをユーザに確認させる（S3）。ユーザが実施済データを削除する旨の指令がキーボード3から入力されたならば（S3：Yes）、削除する実施済データをFDD部5のフロッピーディスク11に保存するかどうかを表すメッセージを表示してユーザに確認させる（S4）。

【0017】実施済データを保存しない旨の指令がキー

10

20

30

40

50

ボード3から入力されたならば(S4:No)、実施済データをRAM14から削除して(S10)処理を終了する。実施済データを保存する旨の指令が入力された場合(S4:Yes)は、FDD部5に備えられているフロッピーディスク11のメモリ残量をチェックする(S5)。

【0018】フロッピーディスク11の空き容量が少なく実施済データを保存できない場合(S5:NG)は、液晶表示装置7に「保存可能なメモリ容量がありません。」等のエラーメッセージを表示して(S13)、データ保存処理をスキップし、実施済データを削除して(S10)処理を終了する。FDD部5のフロッピーディスク11のメモリ残量が十分あり実施済データが保存できる場合(S5:OK)は、ファイルのオープン処理を実行する(S6)。ファイルのオープン処理とは指定したファイル名のファイルをオープンする処理である。例えば実施済データのファイル名を「CHECK」とすると、「CHECK」というファイル名のファイルをオープンする。次に、ファイルID、ファイル管理情報(スケジュールデータ登録件数等)をFDD部5を使ってフロッピーディスク11へ書き込み(S7)、実施済データをフロッピーディスク11へ書き込む(S8)。ファイルに必要な情報を書き込むと、ファイルクローズ処理を実行する(S9)。これはファイルオープン処理(S6)によりオープンされたファイルをクローズする処理を示す。次に、実施済データをRAM14から削除して(S10)、処理を終了する。

【0019】次に、図4のフローチャートを参照しつつ、図3の、実施済データのRAM14から削除する処理(S10)にて、実行される処理手順について説明する。まず、RAM14にスケジュールデータが有るかどうかをチェックする(S20)。スケジュールデータがRAM14に無い場合(S20:無)は、削除すべきデータがないので何もせずにそのまま終了する。スケジュールデータが有る場合(S20:有)は、インデックスとしての記録番号Iに、スケジュールデータの記録番号の最大値をセットする(S21)。次に、Iにセットされた記録番号のスケジュールデータが実施済データであるかどうかをチェック(S22)する。

【0020】実施済データであるかどうかチェックは、既に述べたようにスケジュールデータに不要情報としてVマークあるいはタイムスタンプが付加されているかどうかにより行う。チェックしたスケジュールデータが実施済データでない場合(S22:No)は、スケジュールデータの削除はせずに、Iの値を1だけ減らし、その値で示される記録番号のスケジュールデータをチェックする(S26, S27, S22)。Iが示すスケジュールデータが実施済データである場合(S22:Yes)は、そのスケジュールデータが最終記録であるかどうかをチェックする(S23)。

【0021】最終記録であるということは、Iの値が記録番号の最大値と等しいか否かにより判断される。最終記録である場合(S23:Yes)は、その1レコード分のスケジュールデータが消去される(S25)。Iが示すスケジュールデータが最終記録でない場合(S23:No)は、そのスケジュールデータを消去した後、消去したスケジュールデータ以後のデータのスライド処理を実行する(S24)。スライド処理とは、RAM14内に格納されるスケジュールデータをアドレス順に空きが出ないように格納し直す作業である。即ち、消去したスケジュールデータの格納されていたアドレスに、次のアドレスに格納されているスケジュールデータを格納する。次のアドレスには、その次のアドレスに格納されているスケジュールデータを格納する。この処理を最終記録まで繰り返す。データのスライド処理を実行した後、スライド処理実行前に最終記録となっていたアドレスのスケジュールデータを消去する(S25)。このように1レコード分のスケジュールデータを削除すると、レコード番号Iを一つだけ減らし(S26)、Iがゼロとなるまで処理S22~S26を繰り返し実行する(S27)。以上のようにRAM14から実施済データを削除する処理が実行される。

【0022】尚、第1発明にて、第1記憶手段としてのRAM14から削除する予定情報をフロッピーディスク11へ保存する処理は、図3に示すフローチャートのS4~S10に示す処理である。第2発明にて、記憶手段としてのRAM14に十分な空き容量があるかどうかのチェックは、図3に示すフローチャートのS1に示す処理であり、予め定められた未使用容量(実施例では1レコード分の記憶容量)がない場合に不要情報が付加されたスケジュールデータを出力する処理は、S2およびS3に示す処理である。

【0023】以上説明したように、本実施例の電子タイプライタ1に搭載されたスケジューラによれば、RAM14に格納されている実施済データを削除する際、その実施済データをFDD部5のフロッピーディスク11に保存することを可能としている。そのため削除してしまった実施済データでもFDD部5のフロッピーディスク11から読み出すことができ、ユーザは削除した後でも確認することができる。また特に、誤って必要なスケジュールデータを削除してしまった場合はきわめて有効となる。さらに、CPU10はRAM14のメモリ容量をチェックしており、1レコード分の空き容量がなければ、実施済データを液晶表示装置7に表示して、ユーザに削除するかどうかのチェックを促している。そのためユーザはRAM14の空き容量を気にすることなく、スケジューラを使うことができる。このように実施済みの不要データの管理にてユーザが煩雑な手間を加える必要のない使い勝手のよいスケジューラが提供された。

【0024】また本実施例では、第2記憶手段としての

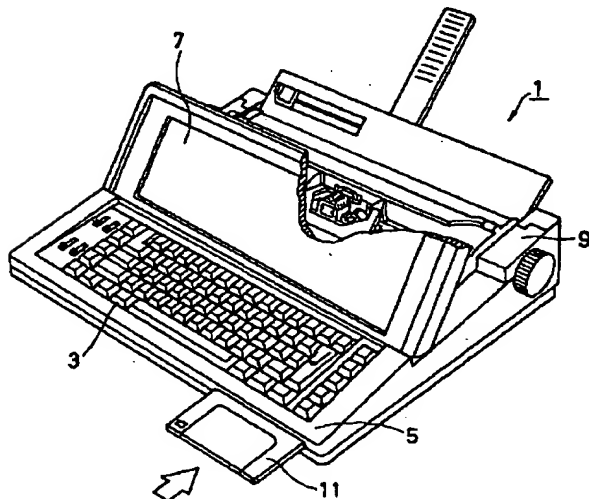
7

フロッピーディスク11のメモリチェックも行っているが、このようにすると前もってフロッピーディスク11にスケジュールデータが保存できるかどうかを確認することができる。またRAM14からスケジュールデータを削除した際、スライド処理を実行することにより、RAM14へのアクセスが速くなる、メモリの使用効率が良いという効果も挙げられる。

【0025】尚本実施例では、スケジューラを電子タイプライタ1に適応したが、これはパーソナルコンピュータ、ワードプロセッサ、電子手帳等さまざまな電子機器に適応させることができる。出力手段としてもディスプレイ装置だけでなく音声出力とすることもできる。第2記憶手段としても、フロッピーディスク11の他に、光磁気ディスク、ハードディスク、磁気テープ等を用いることができる。

【0026】また本実施例では、例えば図3の、実施済データを削除するか(S3)、および削除する実施済データを保存するか(S4)をユーザに決定させる構成としたが、これは予め定められた手順に従ってスケジューラが自動で制御するように構成することもできる。例えば削除する実施済データは自動的にフロッピーディスク11に保存されるようにするとか、実施済データは古いものから自動的に削除される等の制御が考えられる。またRAM14の残り容量のチェックするタイミングも新しくスケジュールデータを入力するときである必要はなく、スケジュールデータを入力したタイミングで実行し

【図1】



8

てもよいし、電源を入れるタイミングで実行してもよい。

## 【0027】

【発明の効果】以上詳述した通り、本発明のスケジューラにより、実施済みの不要データの管理にてユーザが煩雑な手間を加える必要のない使い勝手のよいスケジューラが提供された。

## 【図面の簡単な説明】

【図1】 本発明一実施例としてのスケジューラを搭載した電子タイプライタの全体図である。

【図2】 電子タイプライタの電氣的構成を示すブロック図である。

【図3】 そのCPUが実行する処理を示す第1フローチャートである。

【図4】 そのCPUが実行する処理を示す第2フローチャートである。

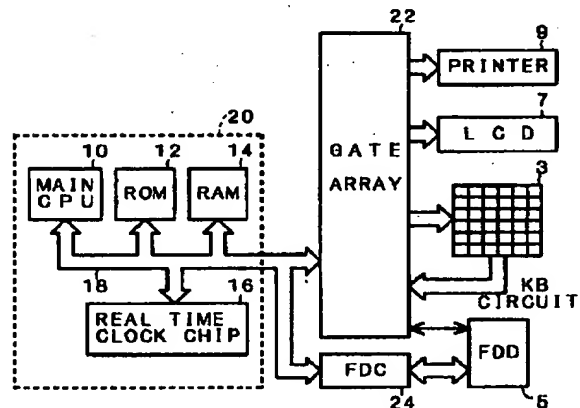
【図5】 従来技術としてのスケジューラの表示部を示す説明図である。

【図6】 本発明の構成例示図である。

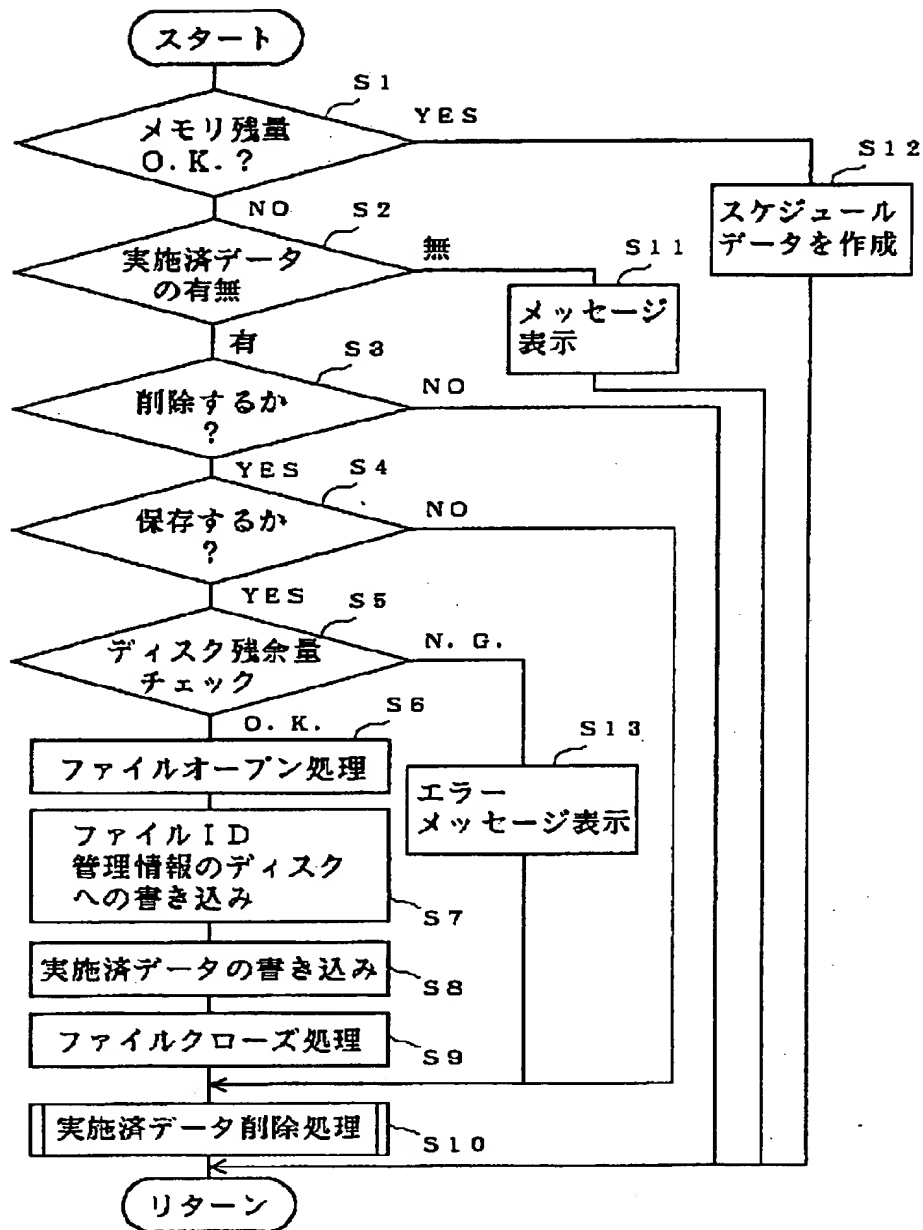
## 【符号の説明】

1・・・電子タイプライタ、3・・・キーボード、  
5・・・FDD部、7・・・液晶表示装置、  
11・・・フロッピーディスク、10・・・CPU、  
12・・・ROM、14・・・RAM、  
18・・・バス、20・・・制御部、  
24・・・FDC

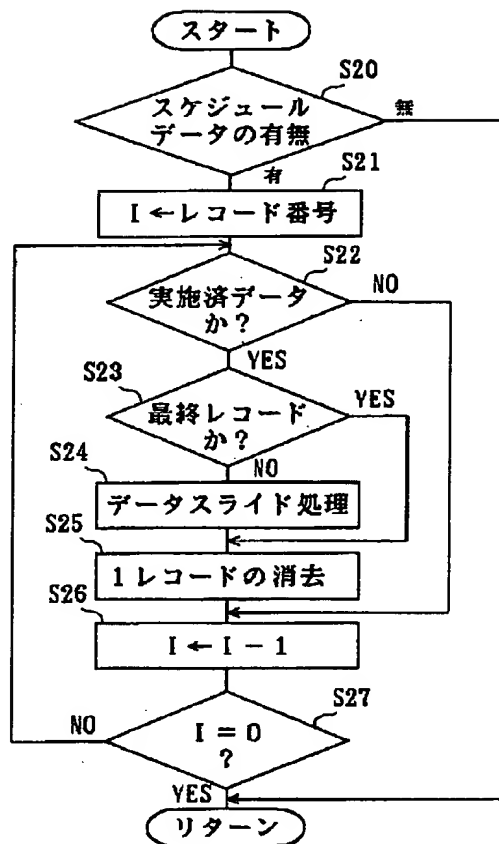
【図2】



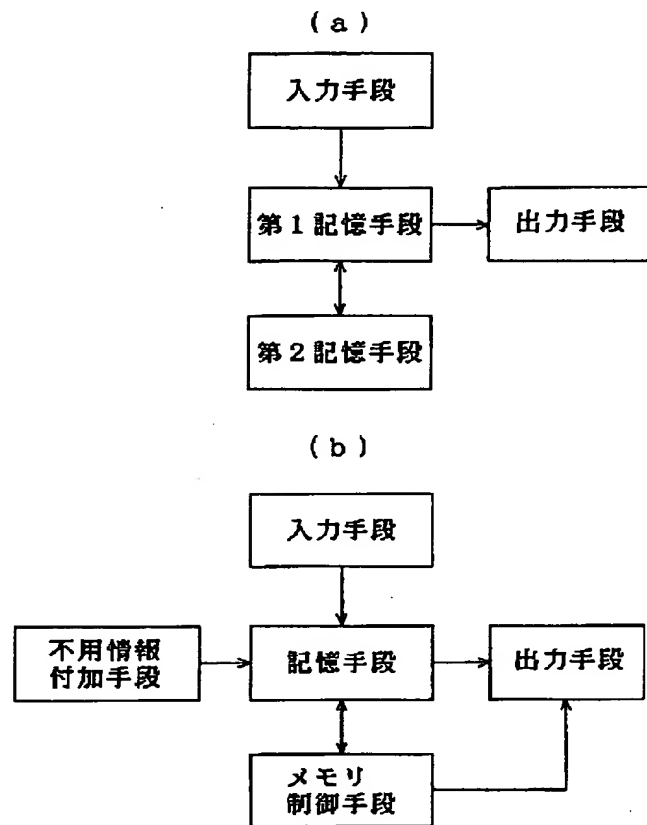
【図3】



【図4】



【図6】



【図5】

| CHECK | TOPIC                 | TO DO                            | DEADLINE |
|-------|-----------------------|----------------------------------|----------|
| ✓     | Stockbroke Management | For meeting this All managem'ent | 8/20 Thu |
|       | Call                  | He would like to                 | 7/20 Sat |
| ✓     | Meeting               | After meeting ti                 | 7/14 Sun |
|       | Pick up               | I will pick up t                 | 8/ 1 Thu |
|       | Party                 | Keving will have                 | 8/31 Sat |
| ✓     | Tennis                | He will play ten                 | 7/ 5 Fri |
|       | Reunion               | A high scholl re                 | 8/20 Tue |
|       | Excursion             | They will make a                 | 8/11 Sun |
|       | Ski                   | I have to make a                 | 8/19 Mon |
|       | Exam                  | She will take an                 | 8/19 Mon |